# Robust Language Recognition via Adaptive Language Factor Extraction

*Brecht Desplanques, Kris Demuynck, Jean-Pierre Martens*

ELIS Multimedia Lab
Ghent University - iMinds, Belgium
brecht.desplanques@ugent.be

## Abstract

This paper presents a technique to adapt an acoustically based language classifier to the background conditions and speaker accents. This adaptation improves language classification on a broad spectrum of TV broadcasts. The core of the system consists of an iVector-based setup in which language and channel variabilities are modeled separately. The subsequent language classifier (the backend) operates on the language factors, i.e. those features in the extracted iVectors that explain the observed language variability. The proposed technique adapts the language variability model to the background conditions and to the speaker accents present in the audio. The effect of the adaptation is evaluated on a 28 hours corpus composed of documentaries and monolingual as well as multilingual broadcast news shows. Consistent improvements in the automatic identification of Flemish (Belgian Dutch), English and French are demonstrated for all broadcast types.

**Index Terms**: language recognition, iVectors, language factor extraction, model adaptation

## 1. Introduction

Belgium has three official languages, namely Dutch (Flemish), French and German. As a consequence, Flemish news broadcasts often encompass French speech segments (German only occurs rarely). Moreover, foreign speech in international news items is not dubbed but presented with subtitles. Other programs, such as documentaries, use a native narrative voice to tell the story, but the sound track also contains foreign language speech fragments that are subtitled or are presented as background while the narrator is telling the story. As a result, language recognition (LR) is an indispensable preprocessor in any computer assisted TV captioning system for a Flemish TV broadcaster.

Several LR techniques have been developed over the last couple of years. They can be regarded as either phonotactic or acoustic. Phonotactic methods (e.g. [1]) are based on the frequencies of short phone sequences in the output of a phone recognizer. Acoustic methods on the other hand directly classify the speech segments on the basis of their acoustic properties.

In this paper we depart from a state-of-the-art acoustic system that employs iVectors [2] to characterize the speech of a speaker. However, contrary to common practice, it does not project all variability in a single Total Variability space, but combines iVectors with Joint Factor Analysis (JFA) [3] in order to separate the language variability from the channel variability [4]. In other words, it extracts language factors as well as channel factors per speaker. Similar to [5] we use the term channel variability to refer to all sources of variability different from the language. Hence, in an LR setup the channel factors encode both channel and speaker variability.

To cope with foreign accents and the everlasting influence of dialect on the standard language we propose to adapt the language factors to the language variants encountered in the audio file that is processed. We also propose modifications to the language classifier (the backend), so that it can process the adapted language factors more effectively.

To evaluate the systems, we compiled a dataset containing monolingual and multilingual news broadcasts as well as a large variety of documentaries. The system is judged on its ability to identify English, Flemish and French speaker segments. The collected dataset exposes a large variety in background conditions and it contains a fair amount of non-native speakers and native speakers with strong accents. Our experiments show reductions of the Speaker Error Rate (SER) by more than 20% relative by adapting the language variability model. Compared to a standard iVector system, the relative improvements can be as large as 50%.

## 2. Baseline system

For this work, we focus on the automatic language recognition per speaker from a closed set of languages. We start from a manual labeling of the speakers and retain only these speech fragments complying with the closed language set for training and evaluation.

### 2.1. Feature extraction, selection and normalization

The acoustic inputs are shifted delta cepstral (SDC) feature vectors [6]. They are acknowledged to constitute a richer representation of the signal dynamics than the standard dynamical features derived from the MFCCs. SDC features are defined by four parameters: $N$, $d$, $P$ and $k$. The first $(2k+1)N$ features of frame $t$ consist of the delta's of the $N$ static MFCCs $c_1 \dots c_N$, computed for frames $t + iP$ ($i = -k \dots k$). A delta at frame $t$ is computed over the window $(t-d, t+d)$. We found $N = 10$, $d = 2$, $P = 3$ and $k = 2$ to perform optimal on broadband data. Supplementing the SDCs with 19 static MFCCs (the $c_1 \dots c_{19}$ of frame $t$) and a normalized log-energy finally leads to a feature vector of dimension 70.

Feature Warping [7] by means of a monotonic non-linear function is applied to achieve a standard normal distribution for each individual feature. The Feature Warping is preceded by a frame selection module which retains mainly frames from the syllable nuclei because these high-energetic frames are the least affected by background noise. The frame selection sometimes retains less than 50% of the frames, but nevertheless improves the LR dramatically, at least for our type of data [4].

### 2.2. iVector extraction

A favored concept in LR is that of iVectors [2] or Total Variability (TV) modeling. All variability is modeled in a single low dimensional subspace. A low rank rectangular matrix $\boldsymbol{T}$, called the TV matrix or the iVector extractor, is used to approximate the GMM mean supervector $\boldsymbol{m}_s$ of speaker $s$ as

$$\boldsymbol{m}_s = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{x}_{s,L_s} \tag{1}$$

with $\boldsymbol{m}$ being the Universal Background Model (UBM) supervector and $\boldsymbol{x}_{s,L_s}$ being the iVector that contains all the information concerning speaker $s$ and language $L_s$ spoken by that speaker. The procedure for extracting iVectors is described in detail in [8].

We determine matrix $\boldsymbol{T}$ by means of Principal Component Analysis (PCA) initialization [9] followed by iterating the non-simplified Expectation-Maximization (EM) algorithm described in [8] until it converges.

### 2.3. Language factor extraction

In (1) the matrix $\boldsymbol{T}$ models both language and channel variability. In order to suppress the effect of the channel variability on the LR, we conceived a method for separating (factorize) these two sources of variability [4].

Let the low rank rectangular matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ be the channel variability matrix and language variability matrix respectively. Then one can model the GMM supervector $\boldsymbol{m}_s$ of speaker $s$ as

$$\boldsymbol{m}_s = \boldsymbol{m} + \boldsymbol{U}\boldsymbol{x}_s + \boldsymbol{V}\boldsymbol{x}_{L_s} \tag{2}$$

with $\boldsymbol{x}_s$ and $\boldsymbol{x}_{L_s}$ being the channel and language factors respectively. These factors can be extracted simultaneously by stacking $\boldsymbol{U}$ and $\boldsymbol{V}$ into one matrix and by following the conventional iVector extraction procedure. The emerging vector $\boldsymbol{x}$ can then be decomposed as $\boldsymbol{x} = [\boldsymbol{x}_s \ ; \ \boldsymbol{x}_{L_s}]$ and all relevant language information is supposed to be included in $\boldsymbol{x}_{L_s}$.

The training of channel variability matrix $\boldsymbol{U}$ is similar to the training of the Total Variability matrix $\boldsymbol{T}$. First we compute the eigenvectors of the within-class (within-language) covariance matrix and subsequently, the EM training is performed. In the EM stage, the first order moments are centered around the ML means of the (annotated) speaker language, as motivated in [3].

Since the number of languages is small, there is no need to rely on the EM algorithm for finding a compact representation $\boldsymbol{V}$ of the language sub-space. Instead, we assign one vector directly to each of the languages and set the values of the column vectors $\boldsymbol{V}_l$ of $\boldsymbol{V}$ equal to the offset between the ML supervector $\boldsymbol{m}_l$ of the corresponding language $l$ (obtained by ML training initialized with the UBM) and the UBM supervector $\boldsymbol{m}$:

$$\boldsymbol{V}_l = \boldsymbol{m}_l - \boldsymbol{m} \tag{3}$$

This ensures that the UBM can be shifted towards the language dependent GMMs when performing adaptation with matrix $\boldsymbol{V}$.

### 2.4. Language classifiers

The backend of the LR system consist of a simple classifier. The input $\boldsymbol{y}_s$ of this classifier for speaker $s$ consists of either the iVectors $\boldsymbol{x}_{s,L_s}$ or the language factors $\boldsymbol{x}_{L_s}$.

#### *2.4.1. Language factor maximum detector (MAX)*

The MAX classifier interprets the language factors $\boldsymbol{x}_{L_s}$ as language scores and simply selects the language producing the highest score.

#### *2.4.2. Gaussian Backend (GB)*

The GB classifier models the distribution of the score vectors $\boldsymbol{y}_s$ of target language $l$ by means of a multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma}$ a full covariance matrix shared by all target languages [10]. The classification is based on the following language score:

$$y^*_{s,l} = (\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_l)^T \boldsymbol{y}_s - \frac{1}{2}\boldsymbol{\mu}_l^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_l \tag{4}$$

#### *2.4.3. Cosine Distance Scoring (CDS)*

The use of Cosine Distance Scoring (CDS) for LR consists of the following steps [11]. First a Linear Discriminant Analysis (LDA) is conducted on the inputs so as to maximize the interclass variability and to minimize the intra-class variability. The resulting vectors are then normalized to have a unit length. Finally, by assuming that each language can be modeled by a von Mises-Fisher distribution, one can retrieve the ML mean of language $l$ from the normalized vectors $\boldsymbol{y}'_s$ as follows:

$$\boldsymbol{\mu}_l = \frac{\sum_{\forall s \in l} \boldsymbol{y}'_s}{\| \sum_{\forall s \in l} \boldsymbol{y}'_s \|} \tag{5}$$

The sum is taken over all speakers $s$ belonging to the training data for language $l$. Assuming identical spreads ($\kappa$) for all language distributions, the language score $y^*_{s,l}$ for a test segment $\boldsymbol{y}'_s$ is computed as follows:

$$y^*_{s,l} = \kappa \boldsymbol{\mu}_l^T \boldsymbol{y}'_s \tag{6}$$

## 3. Proposed methods

### 3.1. CDS in the supervector domain

It seems odd to retrieve the main language directions from (5) as similar information is represented by the supervectors $\boldsymbol{V}_l$ (columns) of the language variability matrix $\boldsymbol{V}$. We therefore propose to construct a classifier by relying solely on information stored in $\boldsymbol{V}$. This also allows us to adapt the classifier by adapting $\boldsymbol{V}$ to the test conditions, as is explained in the next section.

We can realize our goal by calculating the cosine distance based on angles in the language supervector space $L$ defined by the range of $\boldsymbol{V}$ instead of directly interpreting the language factors as coordinates in the Euclidean space $\mathbb{R}^{N_l}$ with $N_l$ the number of languages, as it was done in Section 2.4.3.

Since computing cosine distances is easier in an orthonormal space, we first determine an orthonormal basis $\boldsymbol{Q}$ for the space $L$ via the Gram-Schmidt $QR$ decomposition:

$$\boldsymbol{V} = \boldsymbol{Q}\boldsymbol{R} \tag{7}$$

The coordinates of the supervectors in $\boldsymbol{V}$ with respect to the orthonormal basis $\boldsymbol{Q}$ are given by the $N_l \times N_l$ upper triangular matrix $\boldsymbol{R}$.

During evaluation the language-dependent shift of the UBM supervector for speaker $s$ is given by $\boldsymbol{V}\boldsymbol{x}_{L_s}$. The coordinates $\boldsymbol{y}_s$ of this vector in space $L$ with respect to $\boldsymbol{Q}$ are determined by

$$\boldsymbol{y}_s = \boldsymbol{R}\boldsymbol{x}_{L_s} \tag{8}$$

To compute the CDS in the supervector domain, we normalize the length of $\boldsymbol{y}_s$ and retrieve the language scores from (6). Remember that column $l$ of $\boldsymbol{R}$ contains the coordinates of the language mean $\boldsymbol{V}_l$ in the orthonormal space defined by $\boldsymbol{Q}$, hence the mean vector $\boldsymbol{\mu}_l$ for each language $l$ equals $\frac{\boldsymbol{R}_l}{||\boldsymbol{R}_l||}$.

## 3.2. Adaptive language factor extraction

### 3.2.1. Motivation

The targeted broadcasts include a considerable amount of non-native speakers and speakers with a heavy accent. For instance, one of the documentaries in our evaluation data contains a lot of African French, which has a distinct pronunciation that clearly differs from the French spoken in our training set. We therefore propose a method for adapting the language variability matrix $\boldsymbol{V}$ to the data seen in the audio file that needs to be processed.

### 3.2.2. Implementation

We use the MAP-adaptation framework [12] to adapt the mixture components $\boldsymbol{V}_{l,k}$ of supervector $\boldsymbol{V}_l$ ($k$ indexes one of the Gaussians in the UBM). The updated component $\hat{\boldsymbol{V}}_{l,k}$ is calculated as a weigthed sum between the original $\boldsymbol{V}_{l,k}$ and an ML estimation $E_{l,k}(\boldsymbol{V}_k\boldsymbol{x}_{L_s})$ of this component given the adaptation data:

$$\hat{\boldsymbol{V}}_{l,k} = (1-\alpha_{l,k})\boldsymbol{V}_{l,k} + \alpha_{l,k}E_{l,k}(\boldsymbol{V}_k\boldsymbol{x}_{L_s}) \tag{9}$$

The estimation $E_{l,k}(\boldsymbol{V}_k\boldsymbol{x}_{L_s})$ is computed across all speakers appearing in the test file:

$$E_{l,k}(\boldsymbol{V}_k\boldsymbol{x}_{L_s}) = \frac{\sum_s \hat{P}(l|s)\sum_{t\in s}\gamma_k(t)\boldsymbol{V}_k\boldsymbol{x}_{L_s}}{n_{l,k}} \tag{10}$$

$$n_{l,k} = \sum_s \hat{P}(l|s)\sum_{t\in s}\gamma_m(t) \tag{11}$$

$$\hat{P}(l|s) = \frac{e^{y^*_{s,l}}}{\sum_{j=1}^{N_l} e^{y^*_{s,j}}} \tag{12}$$

As expressed by (3), $\boldsymbol{V}_{l,k}$ equals the ML estimates of the language-dependent shifts of the Gaussian means relative to the UBM origin. Test data specific estimates for each language can be obtained by weighting each speaker in the test file with the posterior probability $\hat{P}(l|s)$ that he speaks language $l$. Note that posteriors $\hat{P}(l|s)$ are retrieved from the language scores emerging from the non-adapted language classifier. As the MAX classifier does not produce log-likelihood scores, we fall back to hard decisions in that case. In cases where we do use $\hat{P}(l|s)$ we set it to zero when it turned out to be smaller than the prior probability $1/N_l$ of language $l$. This suppresses the impact of unreliable decisions.

By including the UBM mixture occupation probabilities $\gamma_m(t)$ in (10), we achieve that estimated shifts $\boldsymbol{V}_k\boldsymbol{x}_{L_s}$ emerging from more reliable language factors (based on more data) have a larger impact on the final outcome. It also means that if there was no evidence for mixture $k$ in the observation data, $\boldsymbol{V}_{l,k}$ will not be updated.

Using the standard MAP procedure one can define the weighting factor $\alpha_{l,k}$ in (9) as

$$\alpha_{l,k} = \frac{n_{l,k}}{n_{l,k} + r_V} \tag{13}$$

The relevance factor $r_V$ determines how strongly the adaptation data can change the original model.

With the newly obtained $\hat{\boldsymbol{V}}_{l,k}$ we can then compute the new language factors $\boldsymbol{x}^*_{L_s}$ by substituting $\boldsymbol{V}$ by $\hat{\boldsymbol{V}}$ in (2) and supply these to the language classifiers.

### 3.2.3. Adaptation of the language classifier

The interpretation of language factors as language scores still holds after adaptation. Hence, the maximum detection classifier can process the new $\boldsymbol{x}^*_{L_s}$ directly.

However, since the language factor distribution may have changed due to the adaptation, the GB classifier should be adapted. To that end, we use the language factors $\boldsymbol{x}^*_{L_s}$ to perform MAP-adaption of the language mean vectors $\boldsymbol{\mu}_l$ of the different languages:

$$\hat{\boldsymbol{\mu}}_l = (1-\alpha_l)\boldsymbol{\mu}_l + \alpha_l E_l(\boldsymbol{x}^*_{L_s}), \quad \alpha_l = \frac{n_l}{n_l + r_{GB}} \tag{14}$$

with $r_{GB}$ being the relevance factor and with

$$n_l = \sum_{\forall s}\hat{P}(l|s)N_f(s) \tag{15}$$

representing the number of frames involved in the adaptation. In this expression, $N_f(s)$ is the number of available frames for speaker $s$ and $\hat{P}(l|s)$ is the weight of these frames in the adaptation. Clearly, the relevance factors $r_{GB}$ and $r_V$ should not be chosen independently. We set $r_{GB} = r_V N_m$ with $N_m$ being the number of mixtures. The language mean vector $E_l(\boldsymbol{x}^*_{L_s})$ of the adaptation data is estimated as

$$E_l(\boldsymbol{x}^*_{L_s}) = \frac{\sum_{\forall s}\hat{P}(l|s)N_f(s)\boldsymbol{x}^*_{L_s}}{n_l} \tag{16}$$

Since the amount of adaptation data may be limited, no adaptation of the shared covariance matrix is considered.

For the CDS classifier it suffices to insert $\hat{\boldsymbol{V}}$ in the algorithms described in Section 3.1.

# 4. Experiments

The experiments assume that every speaker speaks a single language and hence LR is applied on the concatenation of all data assigned to that speaker. Such a concatenation is called a speaker segment and only segments of speakers that are known to speak English, Flemish or French are considered.

## 4.1. Data

### 4.1.1. Training and development data

The Flemish data are taken from the CGN corpus [13]: 23 hours of speech (935 speakers) are used for model training and another 6 hours are used as development data. The English models are trained on 63 hours of speech from the 1996 HUB4 Broadcast News training data (3748 speakers). The remaining 3 hours constitute our development set. We harvested 16 hours of speech from public RTBF podcasts[1] (403 speakers) as French training data and 7 hours as development data.

### 4.1.2. Evaluation data

The investigated techniques are evaluated on a custom dataset composed of three parts. The so-called MONO part consists of 3 hours of monolingual news files per language. The Flemish data is retrieved from Flemish news broadcasts of the Flemish public broadcaster[2] VRT. The English data is the 1997 HUB4 Broadcast News corpus. The French data is extracted from French news radio podcasts[3].

[1]http://www.rtbf.be/radio/podcast

[2]http://www.vrt.be

[3]http://www.rfi.fr

The BN (broadcast news) part consists of 9 hours of news shows of the public as well as the commercial Flemish broadcaster[4]. Flemish accounts for 92% of the speech, English for 5% and French for 3%.

The DOCU part consists of 10 hours of documentaries, broadcasted by the VRT. It holds a completely different language distribution: Flemish 40%, English 22% and French 38%. Details of the speaker distribution of each test set can be found in Table 1.

| | EN | FL | FR | total |
|---|---|---|---|---|
| MONO | 92 | 139 | 92 | 323 |
| BN | 91 | 524 | 45 | 660 |
| DOCU | 53 | 23 | 113 | 189 |

Table 1: Number of speakers per part of the evaluation data.

### 4.2. Evaluation Measures

We compute the Speaker Error Rate (SER) as the percentage of incorrectly classified speaker segments. Since the SER strongly depends on the prior language probabilities, we also introduce the ratio $C_{YX}$:

$$C_{YX} = \frac{I(X;Y)}{H(X)} \tag{17}$$

$I(X,Y)$ is the mutual information between the recognized and the correct language. $H(X)$ is the entropy of the correct language. The probabilities needed for calculating $C_{YX}$ are retrieved from the confusion matrix summarizing the LR results.

Both SER and $C_{XY}$ give equal weight to each speaker, irrespective of how long each speaker was speaking. We therefore also evaluate the systems in function of Frame Error Rate (FER), i.e. the percentage of misclassified frames.

### 4.3. Baseline systems

The performance of five baseline systems is given in the upper section of Tables 2 and 3. The number of UBM mixtures is set to 256. The rank of matrices $\boldsymbol{U}$ and $\boldsymbol{T}$ is 50. Language Factor Extraction (LFE) clearly outperforms the corresponding iVector systems on the MONO and BN data and is competitive on the DOCU data. We also note that the simple MAX classifier is not significantly outperformed by the more complex language classifiers GB and CDS.

| | MONO | BN | DOCU |
|---|---|---|---|
| iVectors + GB | 4.0 | 11.7 | 9.5 |
| iVectors + CDS | 2.5 | 11.8 | 7.9 |
| LFE + MAX | 4.0 | 7.1 | 9.5 |
| LFE + GB | 2.5 | 7.6 | 8.5 |
| LFE + CDS | 1.5 | 7.3 | 9.0 |
| LFE + CDS* | 1.5 | 7.3 | 7.9 |
| Adaptive LFE + MAX | 2.5 | 6.1 | 6.3 |
| Adaptive LFE + GB | 1.2 | 5.9 | 5.3 |
| Adaptive LFE + CDS* | 1.5 | 6.2 | 6.9 |

Table 2: *Speaker Error Rate (%) of the baseline systems and proposed systems on the evaluation set. The lower the better.*

[4]`http://www.vtm.be`

| | MONO | BN | DOCU |
|---|---|---|---|
| iVectors + GB | 86.2 | 55.0 | 66.7 |
| iVectors + CDS | 89.7 | 53.1 | 68.2 |
| LFE + MAX | 85.2 | 67.8 | 65.6 |
| LFE + GB | 90.3 | 64.5 | 68.6 |
| LFE + CDS | 92.6 | 62.7 | 66.8 |
| LFE + CDS* | 93.6 | 63.2 | 69.6 |
| Adaptive LFE + MAX | 89.5 | 67.5 | 74.3 |
| Adaptive LFE + GB | 94.6 | 67.7 | 79.0 |
| Adaptive LFE + CDS* | 93.6 | 65.3 | 72.6 |

Table 3: *Normalized Mutual Information $C_{YX}$ (%) of the baseline systems and proposed systems on the evaluation set. The higher the better.*

### 4.4. Proposed methods

#### *4.4.1. CDS in the supervector domain*

The middle sections of the two tables show that the proposed CDS in the supervector domain (indicated by *) tends to outperform the standard CDS, but given the limited amount of evaluation data, statistical significance could not be proved. Constructing a language classifier based on information in the language variability matrix $\boldsymbol{V}$ shows to be a viable option.

#### *4.4.2. Adaptive language factor extraction*

The effects of adaptive LFE can be found in the lower sections of the tables. Tuning experiments on the development data lead to a CDS spread $\kappa$ of 5 and a relevance factor $r_V = 100$.

Adaptive LFE reduces the SER in all experiments (across datasets and classifiers) which can be attributed to an improved robustness against speaker accents and background noise.

The largest gains are achieved in combination with the GB classifier: the SER is reduced by more than 20% for all datasets and significant improvements are visible in $C_{YX}$ as well. Adaptive LFE with GB classification outperforms the original iVector systems by about 50% relative in terms of SER.

The FER of the top-performing baseline LFE+CDS reduces from 0.3%, 2.7% and 5.4% for MONO, BN and DOCU respectively to 0.1%, 2.1% and 1.1% for our Adaptive LFE with GB classification.

## 5. Conclusions

We presented an adaptive language recognition system for the challenging domain of TV broadcasts. Since we already used a baseline language factor extraction system that separates the Total Variability in the supervectors in a language and a channel part, we could extend it with a method for adapting the language variability model on a file by file basis. This way we can better cope with heavily dialectic or accented speech.

A performance analysis involving three separate classifiers and three very different evaluation sets, shows that the proposed adaptation technique helps in combination with any classifier but especially in combination with a Gaussian-based classifier. The speaker error rate could be reduced by more than 20% relative on all datasets with respect to a baseline that is already 30% better than a standard iVector system using the same classifier.

## 6. References

[1] P. Matějka, L. Burget, P. Schwarz, and J. Černocký, "Brno University of Technology system for NIST 2005 language recognition evaluation," in *Proceedings of Odyssey 2006: The Speaker and Language Recognition Workshop*, 2006, pp. 57–64.

[2] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[3] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[4] B. Desplanques, K. Demuynck, and J.-P. Martens, "Combining joint factor analysis and iVectors for robust language recognition," in *Proceedings of Odyssey 2014: The Speaker and Language Recognition Workshop*, 2014, 8 pp.

[5] V. Hubeika, L. Burget, P. Matějka, and P. Schwarz, "Discriminative training and channel compensation for acoustic language recognition," in *Proc. Interspeech*, 2008, pp. 301–304.

[6] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, and J. R. Deller, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features," in *Proceedings of ICSLP 2002*, 2002, pp. 89–92.

[7] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proceedings of 2001: A Speaker Odyssey, The Speaker Recognition Workshop*, 2001.

[8] O. Glembek, L. Burget, P. Matějka, M. Karafiát, and P. Kenny, "Simplification and optimization of i-vector extraction," in *ICASSP*, 2011, pp. 4516–4519.

[9] L. Burget, P. Matějka, P. Schwarz, O. Glembek, and J. Černocký, "Analysis of feature extraction and channel compensation in GMM speaker recognition system," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 1979–1986, 2007.

[10] D. M. Gonzàlez, O. Plchot, L. Burget, O. Glembek, and P. Matějka, "Language recognition in ivectors space," in *Proc. Interspeech*, 2011, pp. 861–864.

[11] E. Singer, P. Torres-Carrasquillo, D. Reynolds, A. McCree, F. Richardson, N. Dehak, and D. Sturim, "The MITLL NIST LRE 2011 language recognition system," in *Proceedings of Odyssey 2012: The Speaker and Language Recognition Workshop*, 2012, pp. 209–215.

[12] D. A. Reynolds, "Comparison of background normalization methods for text-independent speaker verification," in *Proc. Eurospeech*, 1997, pp. 963–966.

[13] W. Goedertier, S. M. A. Goddijn, and J.-P. Martens, "Orthographic transcription of the spoken dutch corpus," in *Proc. LREC*, 2000, pp. 909–914.